Monitorowanie i zarządzanie logami w systemach rozproszonych.

mgr inż. Jakub Woźniak

Zarządzanie Systemami Rozproszonymi
Instytut Informatyki
Politechnika Poznańska

Wprowadzenie do monitorowania i zarządzania logami

- Monitorowanie i zarządzanie logami to kluczowe procesy w utrzymaniu stabilności i wydajności systemów rozproszonych.
- Monitorowanie obejmuje ciągłe śledzenie stanu aplikacji oraz infrastruktury, pozwalając na szybkie wykrywanie problemów i reagowanie na nie.
- Zarządzanie logami koncentruje się na zbieraniu, przetwarzaniu i analizie logów z różnych źródeł, umożliwiając diagnozowanie problemów, korelację zdarzeń oraz wykrywanie anomalii.
- W nowoczesnych środowiskach DevOps, centralizacja i automatyzacja tych procesów jest kluczowa.

Wyzwania monitorowania w systemach rozproszonych

- Systemy rozproszone charakteryzują się dużą liczbą komponentów, które mogą działać na różnych węzłach i w różnych lokalizacjach. Dynamika zmian, takich jak skalowanie kontenerów czy wdrażanie nowych mikroserwisów, sprawia, że monitorowanie musi być elastyczne i skalowalne.
- Centralizacja danych staje się trudna, ponieważ logi i metryki pochodzą z różnych źródeł. Ważne jest, aby narzędzia monitorujące były w stanie agregować dane z wielu miejsc, a jednocześnie pozwalały na ich analizę w czasie rzeczywistym.

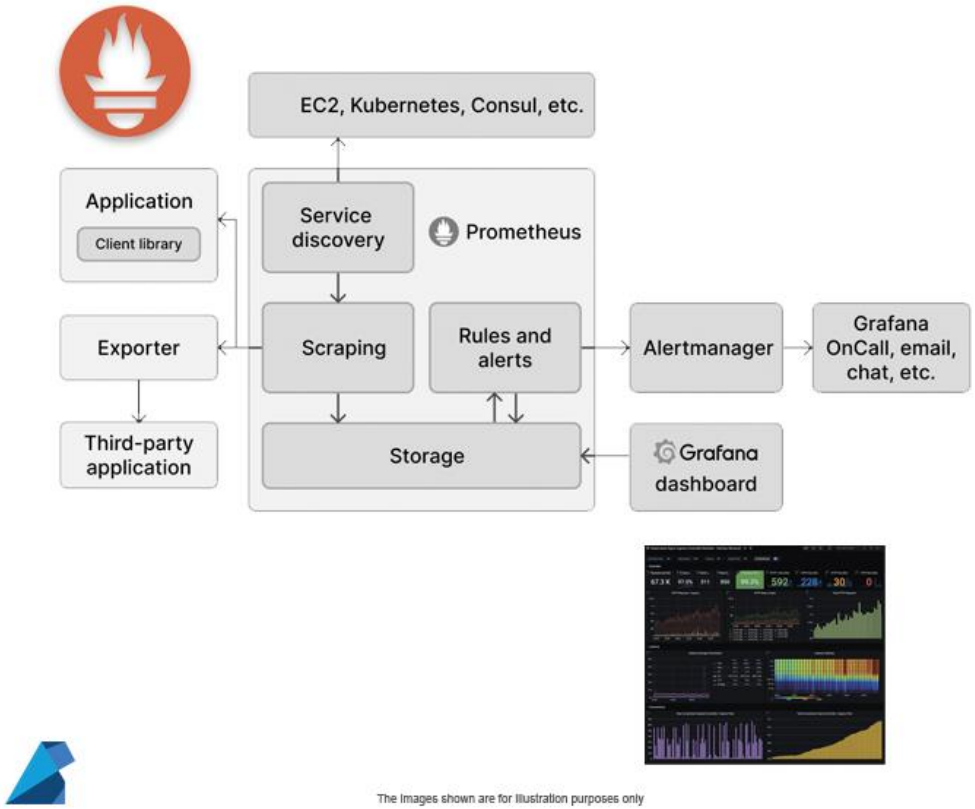
Narzędzia open-source – Prometheus i Grafana

- Prometheus to narzędzie open-source do monitorowania i alertowania, które przechowuje dane jako szeregi czasowe (time-series). Zbiera metryki z aplikacji i infrastruktury za pomocą metody 'scraping'.
- Prometheus jest popularny w środowiskach Kubernetes, ponieważ dobrze integruje się z jego komponentami.
- Grafana służy do wizualizacji danych z Prometheus i innych źródeł, umożliwiając tworzenie interaktywnych dashboardów.
- Dzięki Grafana można śledzić metryki wydajności systemu, analizować trendy i szybko identyfikować problemy.

Architektura Prometheus – jak działa?

- Prometheus działa w oparciu o architekturę pull, gdzie regularnie pobiera metryki (scraping) z różnych źródeł, takich jak eksportery czy aplikacje.
- Dane są przechowywane w lokalnej bazie danych jako szeregi czasowe.
- Użytkownicy mogą definiować reguły alertów, które będą wyzwalane na podstawie określonych wartości metryk.
- Prometheus posiada wsparcie dla eksportowania danych do innych systemów i integracji z Grafana, która umożliwia tworzenie interaktywnych dashboardów do wizualizacji danych.

Architektura Prometheus - przykład



Source: https://medium.com/@tech_18484/understand-prometheus-architecture-1ab83afd53b8

Przykłady eksportera i alertingu w Prometheus

- Exportery to programy zbierające metryki z określonych systemów, np. Node Exporter dla metryk systemu operacyjnego.
- Alerting to system wykrywania problemów na podstawie zebranych metryk, np. wysokie zużycie CPU przez dłuższy czas.
- Przykład konfiguracji alertu: Alert na podstawie liczby błędów HTTP 5xx przekraczającej ustalony próg.

Typy metryk w Prometheus – jakie są i jak ich używać?

W Prometheus istnieją trzy główne typy metryk:

1. Licznik (Counter): Metryka monotoniczna, która zawsze rośnie – np. liczba obsłużonych żądań HTTP.

2. Histogram: Umożliwia śledzenie rozkładu wartości, takich jak czas odpowiedzi serwera, dzieląc je na 'wiadra'.

3. Gauge: Metryka zmieniająca się w czasie, np. aktualne zużycie pamięci lub obciążenie procesora.

Przykład konfiguracji dashboardu w Grafana

- Aby skonfigurować Grafana do wizualizacji danych z Prometheus, należy:
  - Dodać Prometheus jako źródło danych.
  - Utworzyć panele przedstawiające kluczowe metryki, takie jak użycie CPU, pamięci oraz ruch sieciowy.
  - Zdefiniować filtry i wykresy, aby lepiej analizować dane. Można dodać alerty, które będą wyzwalane, gdy określone wartości metryk zostaną przekroczone.

node 10.21.37.1

System

MikroTik

Syst...
3 months

Tem...

62 °C

Rout...
hEX S

Syst...
7.15.2 (stable)

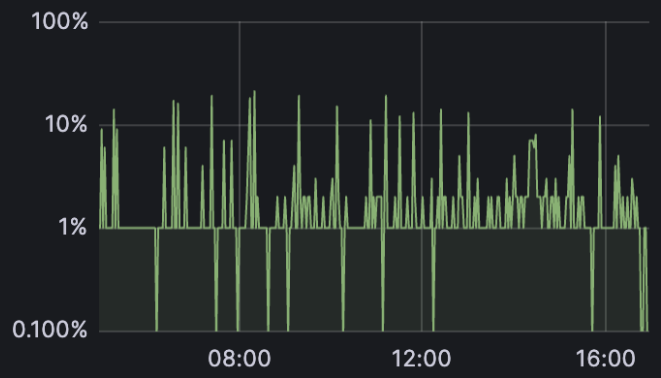
Volt...
50 V

Used RAM...
26.3%

CPU Load
1.0%

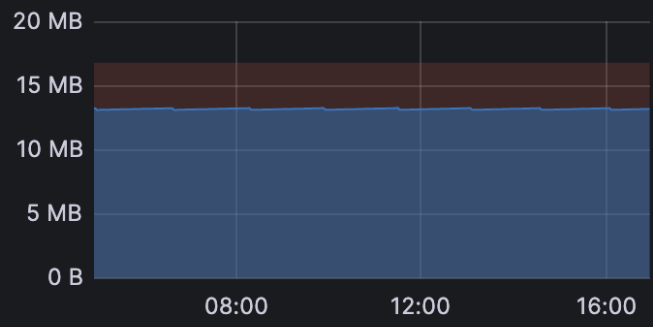
HDD Utiliz...
78.6%

CPU load



	min	max	avg	current
MIPS 1004Kc V2.15	0%	21%	2.11%	0%

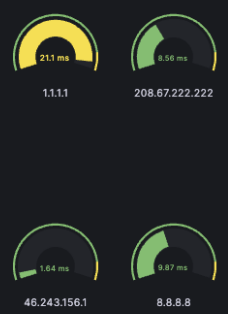
HDD Utilization



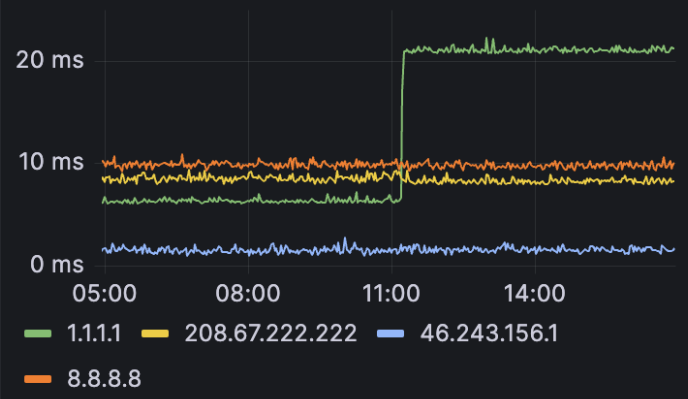
	min	max	avg	curre
Used memory	13.1 MB	13.3 MB	13.2 MB	13.2
Total memory	16.8 MB	16.8 MB	16.8 MB	16.8

Network statistics

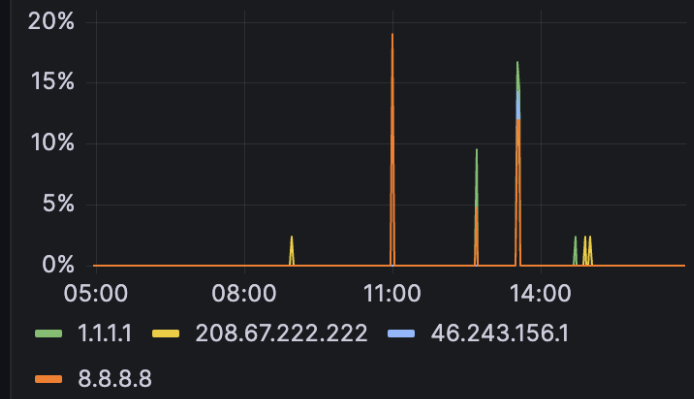
Average ping



Average ping



Average loss



ELK Stack – jak to działa?

- ELK Stack składa się z trzech komponentów:
  - Elasticsearch: Baza danych przeznaczona do składowania i wyszukiwania danych. Indeksuje logi, umożliwiając szybkie wyszukiwanie.
  - Logstash: Narzędzie do przetwarzania logów. Odbiera logi z różnych źródeł, parsuje je i przekazuje do Elasticsearch.
  - Kibana: Platforma do wizualizacji danych z Elasticsearch. Umożliwia tworzenie interaktywnych dashboardów, przeszukiwanie logów i analizę zdarzeń.

Rola Logstash i przykład przetwarzania logów

- Logstash odbiera logi z różnych źródeł, takich jak aplikacje, serwery i pliki. Przetwarza dane, np. parsując je i przekształcając formaty.
- Przykładem może być konwersja logów z formatu JSON na standard logów Apache.
- Dane są następnie wysyłane do Elasticsearch, gdzie są indeksowane.

Fluentd – alternatywa i uzupełnienie ELK Stack

- Fluentd to wszechstronne narzędzie do agregacji logów, które może współpracować z ELK Stack lub działać samodzielnie.
- Zbiera logi z różnych źródeł, przekształca je i przekazuje do systemów takich jak Elasticsearch, InfluxDB czy chmura.
- Dzięki wtyczkom, Fluentd umożliwia przetwarzanie logów w czasie rzeczywistym, filtrowanie, wzbogacanie danych oraz wysyłanie ich do różnych lokalizacji.

Analiza logów w Kibana – co możemy monitorować?

- Kibana pozwala na przeglądanie logów w czasie rzeczywistym, tworzenie interaktywnych dashboardów i definiowanie alertów.
- Użytkownicy mogą monitorować różne aspekty działania aplikacji, takie jak wskaźniki wydajności, bezpieczeństwo oraz wykrywanie anomalii.
- Dzięki funkcjom takim jak przeszukiwanie pełnotekstowe i agregacja danych, Kibana umożliwia szybkie diagnozowanie problemów.

Integracja narzędzi monitorujących z systemami automatyzacji

- Monitorowanie oraz zarządzanie logami można zintegrować z narzędziami automatyzacji, np. Ansible lub Terraform, aby automatycznie reagować na określone zdarzenia i aktualizować konfigurację systemu.
- Przykładem może być restart usługi lub skalowanie systemu po przekroczeniu pewnych progów metryk.
- Takie podejście nazywamy „zero touch” i jest fundamentem współczesnych praktyk DevOps.

Podsumowanie wykładu

- Monitorowanie i analiza logów są kluczowe dla stabilności systemów rozproszonych. Należy stosować odpowiednie narzędzia (Prometheus, Grafana, ELK, Fluentd) oraz integrować je z systemami automatyzacji. Centralizacja logów ułatwia ich przetwarzanie, korelację zdarzeń i szybkie diagnozowanie problemów.
- Istnieje wiele komercyjnych rozwiązań do monitorowania, które z powodzeniem zastępują ich open-sourcowe odpowiedniki. Między innymi są to: Datadog, NewRelic, Splunk, SumoLogic, etc.
- Wraz z upowszechnieniem się technologii opartych o AI (LLM) dotychczasowe podejście wzbogaca się o analizę logów przy pomocy tych modeli. Nazywamy to AIOps